

Proyecto CFGS



CFGS Automatización y Robótica Industrial

“Robot Replicador mediante servomotores”

Autores:

Javier Coronel Vides

Karim Suleiman Fernández

En Andújar (Jaén), 21 de Junio de 2019

Índice

Agracedimientos	7
Resumen	9
Abstract	11
1 - Introducción: Industria 4.0 y Robótica Industrial	13
1.1 Grados de Libertad	16
1.2 Tipos de Robot	17
1.3 Precisión	19
2 – Pliego de Condiciones	20
2.1 Javier Coronel Vides	20
2.2 Karim Suleiman Fernández	20
2.3 Puntos en común	20
3 – Prototipos	22
4 – Presentación y Descripción	27
4.1 PWM	27
4.2 Diagrama de Flujo	29
5 – Proceso de impresión 3D	32
6 – Presupuesto	35
7 – Planos	36
8 – Contratiempos	38
9 – Bibliografía	41
ANEXO i: Tabla de I/O y Código	42
ANEXO ii: Planos acotados	48

Índice de Figuras

- Figura 1: Etapas de la Industria	14
- Figura 2: Partes anatómicas brazo robótico - grados de libertad (g.d.l.)	16
- Figura 3: Partes anatómicas muñeca	16
- Figura 4: Estructura mecánica cartesiana	17
- Figura 5: Estructura mecánica cilíndrica	18
- Figura 6: Estructura mecánica esférica o polar	18
- Figura 7: Estructura mecánica angular o antropomórfica	19
- Figura 8: Estructura mecánica scara	20
- Figura 9: Modelo 0	23
- Figura 10: Modelo 1	24
- Figura 11: Modelo 1.2	24
- Figura 12: Modelo 2	25
- Figura 13: Modelo 2.1	25
- Figura 14: Modelo 2.2	26
- Figura 15: Modelo 2.2 – Acotado	26
- Figura 16: Base	27
- Figura 17: Pulse Width Modulation (Modulación de Ancho de Pulso)	28
- Figura 18: Software de diseño 3D “SketchUP Pro 2019”	33
- Figura 19: Software de depuración diseño 3D “Slic3r”	34
- Figura 20: Software de impresión 3D “Ultimaker Cura 3.2.1”	34
- Figura 21: Proceso de impresión 3D	35
- Figura 22: Presupuesto	36
- Figura 23: Cableado I/O	37
- Figura 24: Diseño Final	38
- Figura 25: Diseño Final	38
- Figura 26: Fallos de impresión del Modelo 2.2	39
- Figura 27: Fallos de impresión brazo/pinza	41
- Figura 28 y 29: Cotas Base y Brazos	48
- Figura 29: Cotas Brida	49
- Figura 30: Cotas entre ejes	49

Agradecimientos

A Rafael Antonio Quesada Calahorro, Ruslan Khatamtsov y Juan López Méndez

Por todos los momentos, el aprendizaje y el tesón recibidos durante estos 2 largos años.

Gracias.

RESUMEN

En el presente proyecto se expone el diseño, fabricación y programación de un brazo robótico basado en la famosa placa de desarrollo Arduino. Las ideas y conceptos integrados en el proyecto han favorecido la adquisición de diferentes destrezas y conocimientos, como pueden ser el dibujo técnico, la impresión 3D, la familiarización e interpretación de código y la integración de información para dar funcionalidad y hacer viable el proyecto. El lenguaje escogido para el desarrollo del mismo ha sido "C". Todo el proceso ha sido documentado para su posterior replicación, aportando los elementos necesarios para ello (materiales, diagramas, métrica y código).

Palabras clave: Brazo robótico, Arduino, diseño, impresión 3d, programación, "C".

ABSTRACT

The present project showed design, making and programming of a robotic arm based in the Arduino board. The ideas and concepts which were be integrated in the Project help us to adquire diferents skills like technical drawing, 3D print, and knowledge and interpretation of code and finally the integration of information to give funtionality to become real the Project. "C" is the language choosen by us to develop the idea. All the process had been documented for future replication, giving the main elements to do it (materials, diagrams, code and metrics)

Keywords: Robot arm, Arduino, Design, 3d print, programming, "C".

1 – Introducción: Industria 4.0 y Robótica Industrial

Durante los últimos años se ha podido observar una mayor polivalencia a la hora de desempeñar un puesto de trabajo técnico. Cada vez más se demanda un perfil de trabajador que domine diversas áreas de conocimiento, y esto conlleva una continua formación. Este ciclo perenne de aprendizaje viene propiciado por un punto de inflexión que está sufriendo la industria desde hace una serie de años: la denominada “Industria 4.0”.

La primera Revolución Industrial, entre los siglos XVIII y XIX, transformó la economía agraria y artesanal mediante la mecanización de procesos de producción. La segunda transición, en el siglo XX, trajo la producción en cadena, con la aparición de fábricas y líneas de montaje que permitieron fabricar productos para el gran consumo. Ya a finales del siglo XX el despliegue de la electrónica y la informática hicieron que los procesos industriales se pudieran automatizar, reemplazando personas por máquinas. Finalmente, y llegando a nuestros días, gracias a la tecnología de Internet, y al desarrollo en las tecnologías como la robótica o los sensores, tenemos como resultado la Industria 4.0.

El término de Industria 4.0, se acuñó en Alemania, y en definitiva se enfoca en la aplicación a la industria del modelo “Internet de las cosas” (IoT) (Davies, R., 2015)

Es por ello por lo que el trabajador debe adquirir competencias que le permitan adaptarse a las nuevas tecnologías que surgen día a día y pueda desempeñar su trabajo de una forma descentralizada, teniendo un carácter de “todoterreno” tan demandada en la Industria del S.XXI. En muchas ocasiones el trabajador adquirirá conocimientos transversales que le permitirán moverse con soltura e interpretar información proveniente de diferentes áreas y departamentos, ganando flexibilidad, eficacia y ahorrando tiempo y dinero a la empresa.

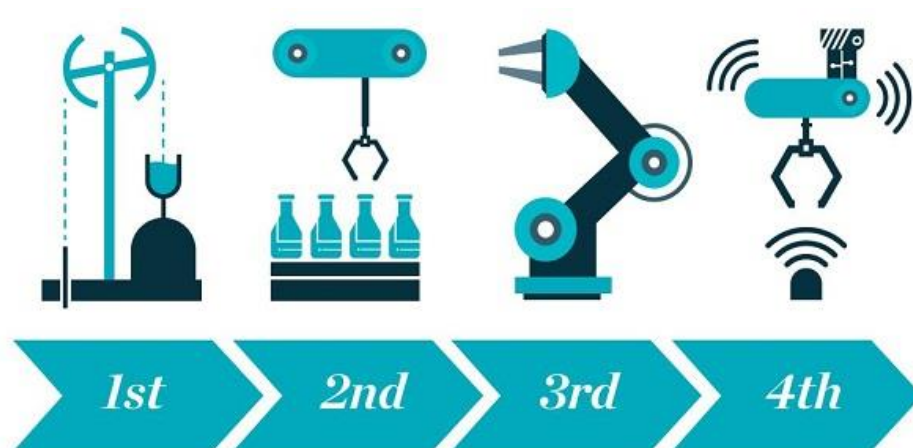


Figura 1: Etapas de la Industria

Estos conocimientos transversales vienen propiciados por la cada vez más débil frontera entre los departamentos de una empresa. Gracias a sistemas de intranet empresariales (véase SAP), las áreas de mantenimiento, producción, diseño o logística están a un “click” de distancia. Valores de producción, ventas, gestión de alertas en mantenimiento, control de stock de almacén...todo está integrado mediante software específico que nos facilita el trabajo.

El trabajador recibe a diario una amplia cantidad de información de diversa procedencia, y es por ello por lo que debe de estar preparado para poder operar entre líneas y hacer frente a la demanda.

Esta revolución tiene un gran efecto multiplicador: impulsa la innovación tecnológica y genera puestos de trabajo cualificado, estable y bien remunerado, por lo que tiene un impacto social inmediato tanto a nivel nacional como internacional (del Val Román, J.L., 2016).

Sin embargo, en el siglo XXI el peso de la industria ha disminuido en Europa, propiciado por la gran crisis económica y la externalización de la producción a países del sudeste asiático. La Unión Europea (UE) se propone aumentar la aportación de la industria al crecimiento, desde el nivel actual del 15,6 % del PIB de la UE, hasta el 20 % en 2020 (Comisión Europea, 2014).

Propiciado por la situación actual del ámbito industrial, en claro crecimiento, se ha decidido la elaboración del presente proyecto. Conlleva labores de diseño, fabricación y programación de un sistema robotizado, adquiriendo competencias transversales que nos aportará una mayor perspectiva del mundo laboral. También se decidió elaborar el proyecto de forma conjunta, permitiéndonos mayor rango de acción y enriquecimiento mutuo, aportando ideas y apoyándonos en las fortalezas de cada uno para dar forma al proyecto.

Como se ha comentado anteriormente, muy ligada a la Industria 4.0 tenemos a la robótica, área multidisciplinar donde las haya, ya que implica tener conocimientos tanto de programación, como de electrónica, electricidad, mecánica incluso de diseño.

Existen multitud de tipos de robots, desde los robots meramente industriales, pasando los que se emplean en el ámbito clínico, de carácter militar o simplemente como servicio. Su empleo en la industria evita que el ser humano desarrolle su labor en ambientes peligrosos o nocivos (química, radiación, refinerías, altos hornos...) y además proporciona la capacidad de poder replicar cualquier movimiento de forma continuada por largos periodos de tiempo, lo que maximiza la producción.

- 1.1 Grados de Libertad:

Se denomina grado de libertad (abreviado, g.d.l.) a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos). También se puede definir los g.d.l. como posibles movimientos básicos (giratorios y de desplazamiento). Normalmente, en cadenas cinemáticas abiertas, cada par eslabón-articulación tiene un solo grado de libertad, ya sea de rotación o de traslación (Miranda Peña, F., 2016).

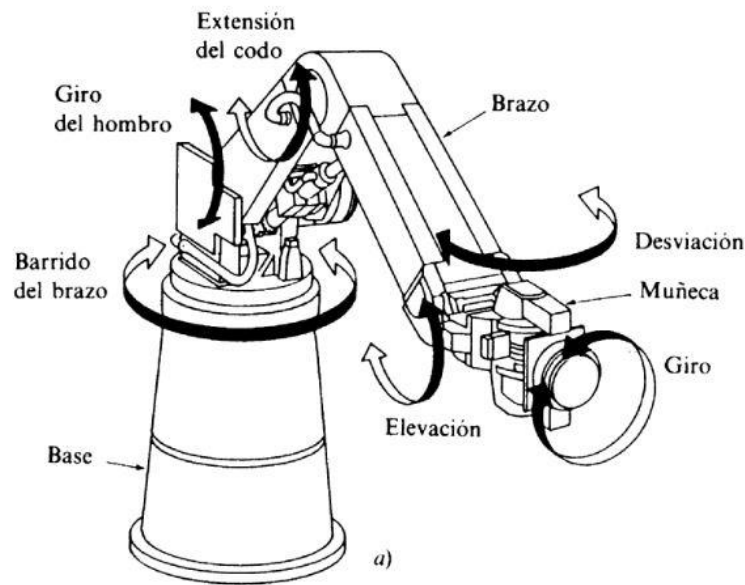


Figura 2: Partes anatómicas brazo robótico, se pueden observar los grados de libertad (g.d.l.)

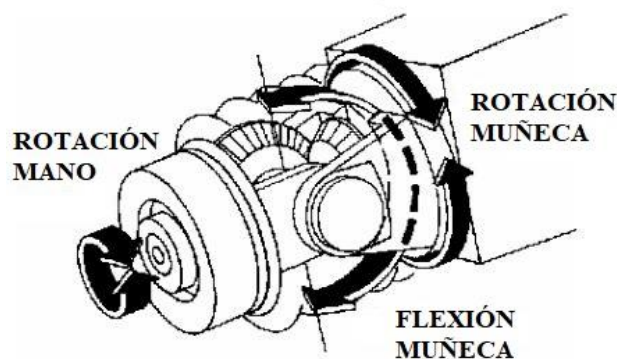


Figura 3: Partes anatómicas muñeca

- 1.2 Tipos de Robots.

Se pueden clasificar los tipos de robots según su construcción o capacidad de movimiento, y son las siguientes (Romeo, A., 2013):

- Cartesiano:

En esta categoría podemos encontrar 2 tipos: de pórtico o rectilíneo. Su volumen de trabajo teórico es L^3 con capacidad de carga independiente de la configuración, precisión y resolución constantes en el volumen de trabajo. Sus principales aplicaciones son el paletizado y la alimentación de máquinas. Tiene una geometría sencilla, con 3 g.d.l., los cuales corresponden con los ejes X, Y, Z.

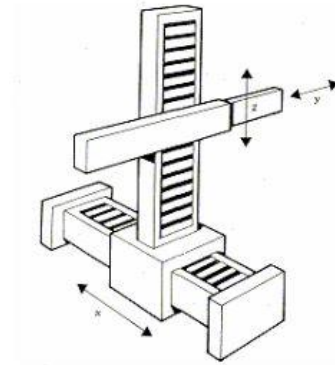


Figura 4: Estructura mecánica CARTESIANA

- Cilíndrica

Su volumen de trabajo teórico es $9L^3$, con capacidad de carga independiente de la configuración, precisión y resolución variables en el volumen de trabajo. Sus principales aplicaciones son la carga y descarga, como la alimentación de máquinas. Presenta 3 g.d.l., siendo dos lineales y uno rotacional.

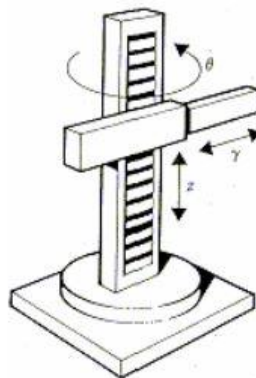


Figura 5: Estructura mecánica CILINDRICA

- Esférica o Polar:

Volumen de trabajo teórico $29L^3$, su capacidad de carga es dependiente de la configuración, con precisión y resolución variables en el volumen de trabajo. Sus aplicaciones son la carga y descarga, la alimentación de máquinas, la soldadura, el montaje, etc. Presenta 3 g.d.l., siendo uno lineal, otro rotacional y otro angular.

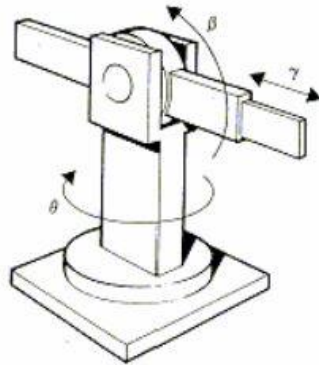


Figura 6: Estructura mecánica ESFÉRICA o POLAR

- Angular o Antropomórfica.

Volumen de trabajo teórico $33L^3$, su capacidad de carga es dependiente de la configuración, con precisión y resolución variables en el volumen de trabajo, de carácter esférico. Sus aplicaciones son muy diversas ya que tiene mejor accesibilidad de trabajo en comparación con las estructuras anteriormente citadas. Presenta 3 g.d.l., teniendo una articulación con movimiento rotacional y dos angulares.

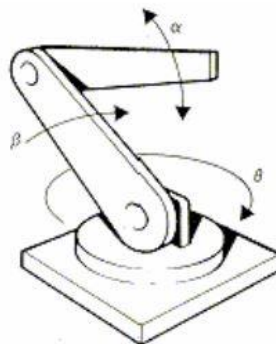


Figura 7: Estructura mecánica ANGULAR o ANTROPOMÓRFICA

- Scara (*Selective Compliance Assembly Robot Arm*).

Volumen de trabajo teórico $12.5L^3$, su capacidad de carga es independiente de la configuración, con precisión y resolución variables en el volumen de trabajo. Sus aplicaciones son principalmente el montaje de precisión. Presenta 3 g.d.l., teniendo dos articulaciones rotacionales y una última lineal.

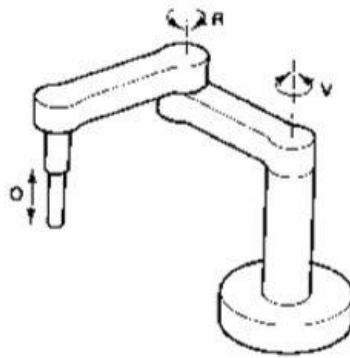


Figura 8: Estructura mecánica SCARA

- 1.3 Precisión

Conceptos como la resolución espacial, exactitud y repetibilidad son factores que intervienen en la precisión del movimiento del robot.

Resolución espacial. Incremento más pequeño de movimiento en que el robot puede dividir su volumen de trabajo

Exactitud. Se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del volumen de trabajo.

Repetibilidad. Se refiere a la capacidad del robot de regresar al punto programado las veces que sean necesarias.

2- Pliego de Condiciones

- 2.1 Javier Coronel Vides

Decidí comenzar este proyecto por el hecho de que supusiese un reto y aprendizaje en un área que no hemos desarrollado durante el curso, que es la robótica. Me interesaba que hubiese la posibilidad de diseñar en entorno 3D y su posterior impresión, ya que sentía curiosidad y ganas de embarcarme en un proyecto que integrase ambos aspectos.

- 2.2 Karim Suleiman Fernández

La decisión de optar por este proyecto y no otro es fundamentalmente el interés por el mundo de las prótesis robotizadas y la terapia ocupacional. Formarme alrededor de esta idea y desarrollar un prototipo funcional básico que me proporcione conocimiento al respecto. Finalmente tener la oportunidad de darle un punto de vista más “humano” a todo el entresijo técnico que conlleva la elaboración del proyecto.

Quise integrar el componente de diseño y no optar por opciones prediseñadas para retarme personalmente y saber hasta dónde puedo llegar a la hora de crear y poner a prueba mi capacidad en esta área.

La imposición de un reto a la hora de desarrollar el proyecto para así adquirir de una forma autónoma y autodidacta competencias que me ayuden a encarar una posible situación real en el mundo laboral. Salir de la zona de confort y de la programación PLC, la cual se ha desarrollado durante la formación en centro de trabajo.

Me gustaría poder llegar a desarrollar una prótesis funcional en un futuro, o incluso dedicarme a ello, siendo este, bajo mi punto de vista, un humilde pero buen punto de partida.

- 2.3 Puntos en común

Ambas partes acordamos desde un primer momento la utilización de la placa de desarrollo Arduino, y “C” como lenguaje de programación, frente a otras placas

como pueden ser RaspberryPi y Python o como lenguaje, debido a la gran comunidad que respalda a ambas (Arduino y C), pudiendo consultar vía internet recursos que nos facilitasen el desarrollo y la resolución de dudas. Su versatilidad nos ayudó a cambiar varias veces el enfoque del proyecto una vez avanzado el mismo.

3– Prototipos

Se ha visto oportuno adjuntar los modelos previos al diseño final, ya que, gran parte del trabajo se ha centrado en la labor de modelaje 3d. Con ello se pretende mostrar el tiempo dedicado al mismo y las dificultades encontradas a la hora de trabajar, tanto en diseño como en la impresión 3d, los cuales eran totalmente nuevos para ambos.

Los adjuntados a continuación son modelos desechados, bien por no adaptarse a las características de la impresora en cuestión (trazo, ángulo de la pieza, etc.) o por no cumplir las expectativas una vez impresos (fallo de cotas, material rígido, fallo en la impresión, etc.).

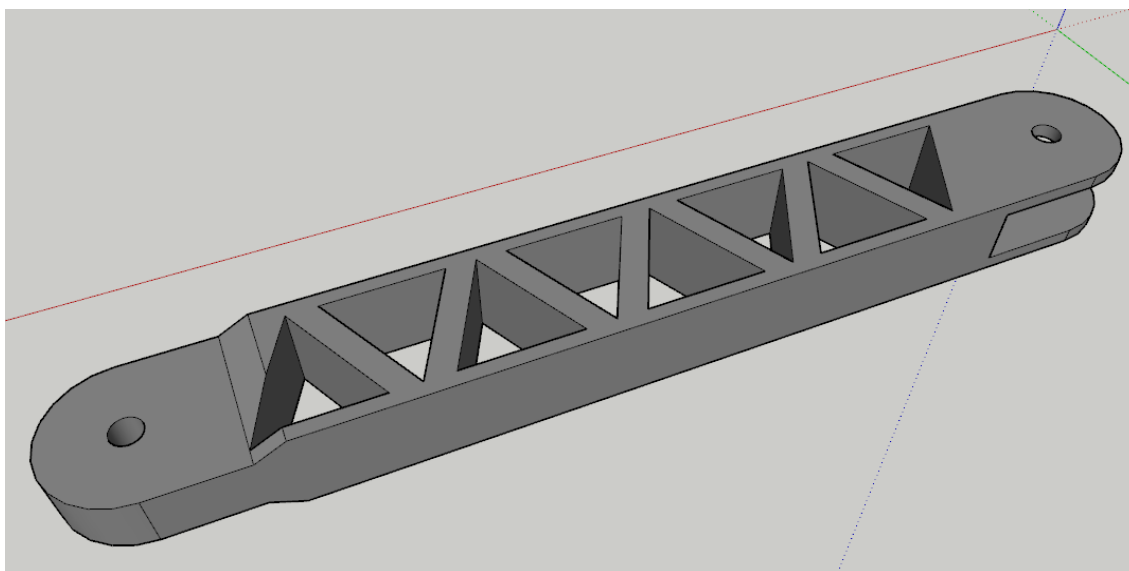


Figura 9. Modelo 0.

Modelo inicial, aun no se tenía claro la funcionalidad del brazo, ni el modelo de los servomotores. Toma de contacto con el software de diseño 3d SketchUp. En este paso se descartaron muchos modelos, a la vez que se puso a prueba la visión y viabilidad del proyecto.

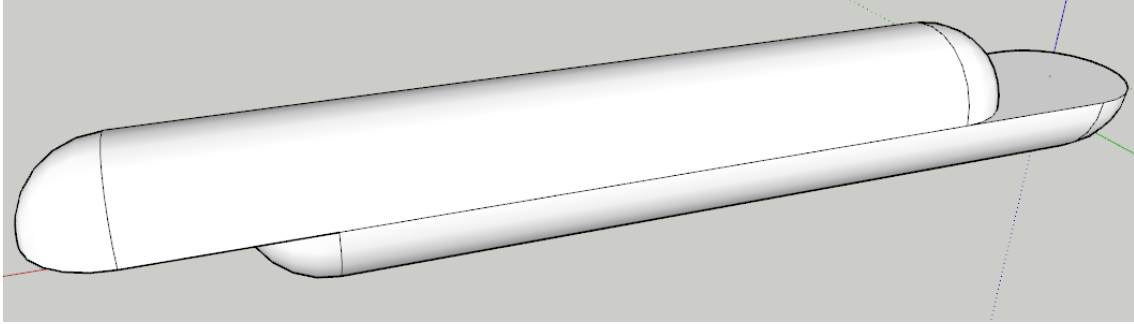


Figura 10. Modelo 1.

Evolución del modelo inicial, se decidió darle un aspecto más moderno, con unas líneas redondeadas. Más tarde (en Modelo 2.2) se podría observar que la primera impresora 3d que utilizamos tenía limitaciones técnicas para hacer frente a este tipo de diseños.

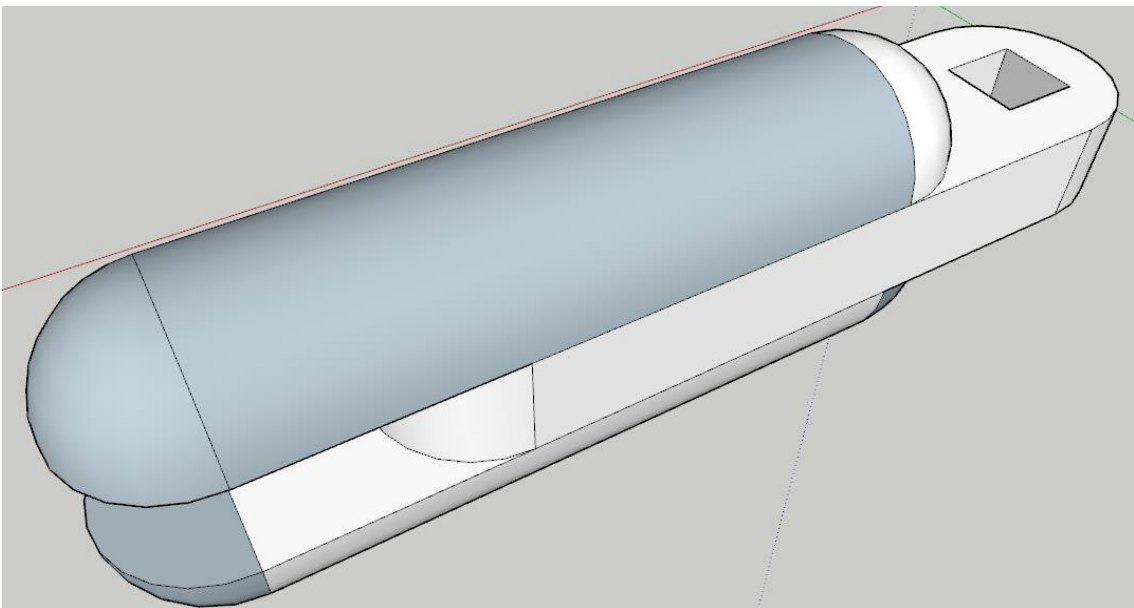


Figura 11. Modelo 1.2.

Se añadió un eje central sobre el que apoyar los servos, se comenzaron a barajar diversos tipos de servomotores. Para este modelo se decidió implementar el servomotor más común del mercado, el SG90.

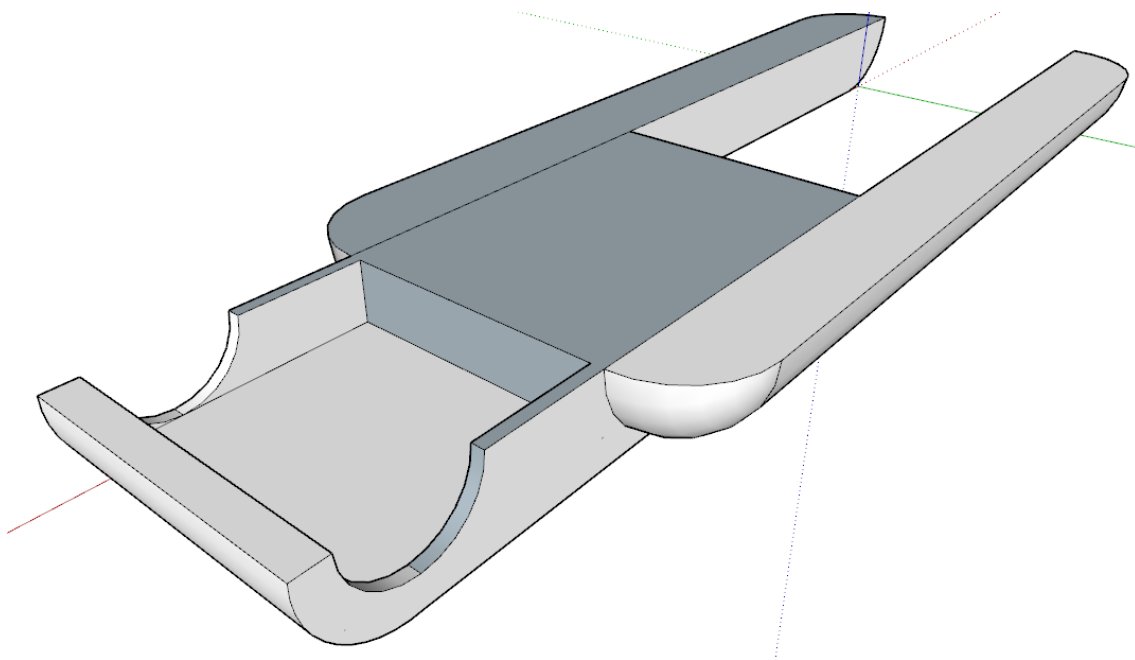


Figura 12. Modelo 2.

Llegados a este punto, se decidió cambiar el diseño por uno más funcional y que basase su estructura en la unión de varias piezas. Observando la envergadura de la estructura, se decidió prescindir de los servomotores SG90 para colocar los servomotores RDS3115MG, de mayor tamaño y fuerza.

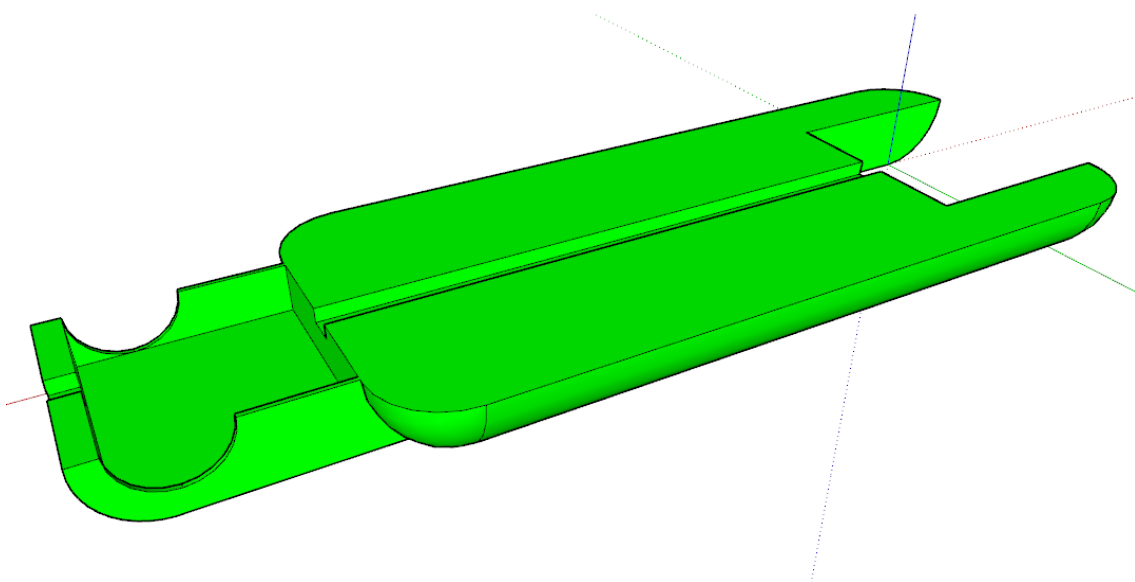


Figura 13. Modelo 2.1.

Inclusión de raíl central para la ubicación del cableado. Se aumentó la zona central para ganar estabilidad en el conjunto.

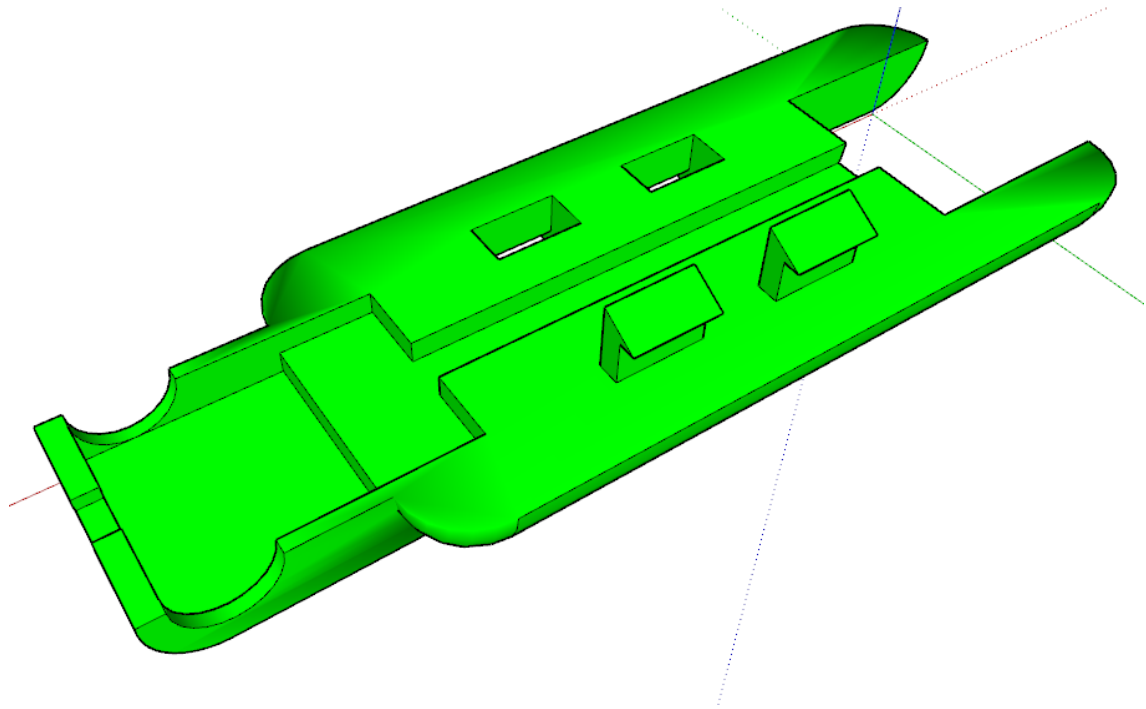


Figura 14. Modelo 2.2.

Modelo final de esta versión, inclusión de sistema de clips que permitiesen el anclaje entre piezas sin ningún tipo de adhesivo.

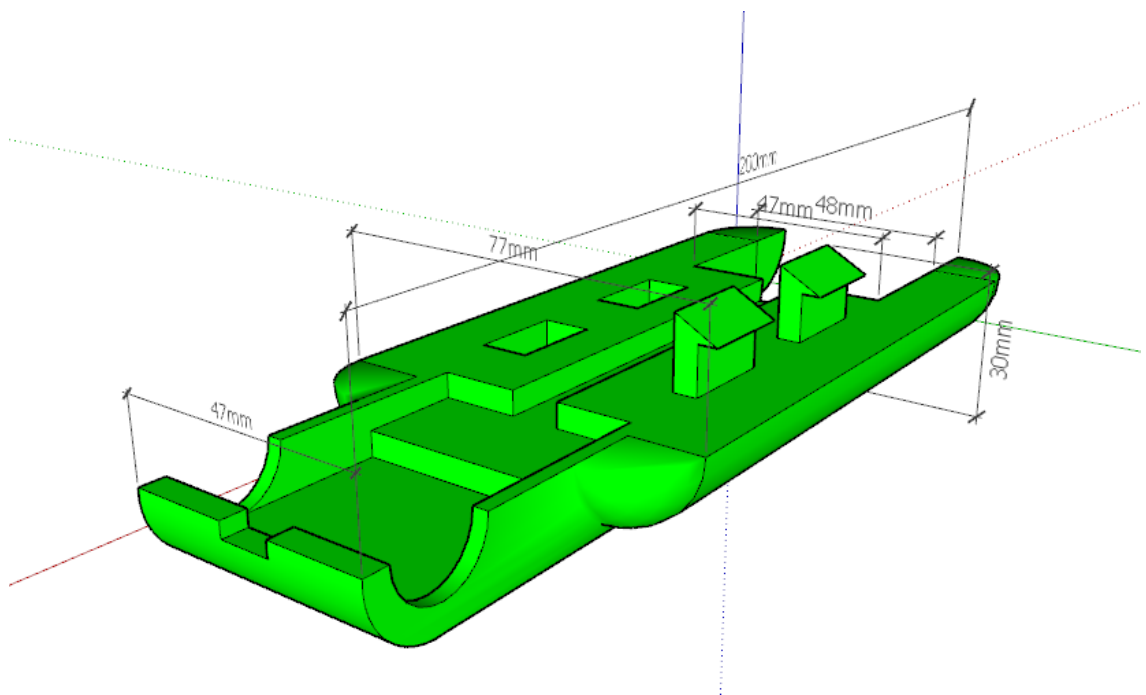


Figura 15. Modelo 2.2 - Acotado.

Una vez recibidos los servomotores RDS3115MG, se establecieron las cotas correctas para su posterior impresión.

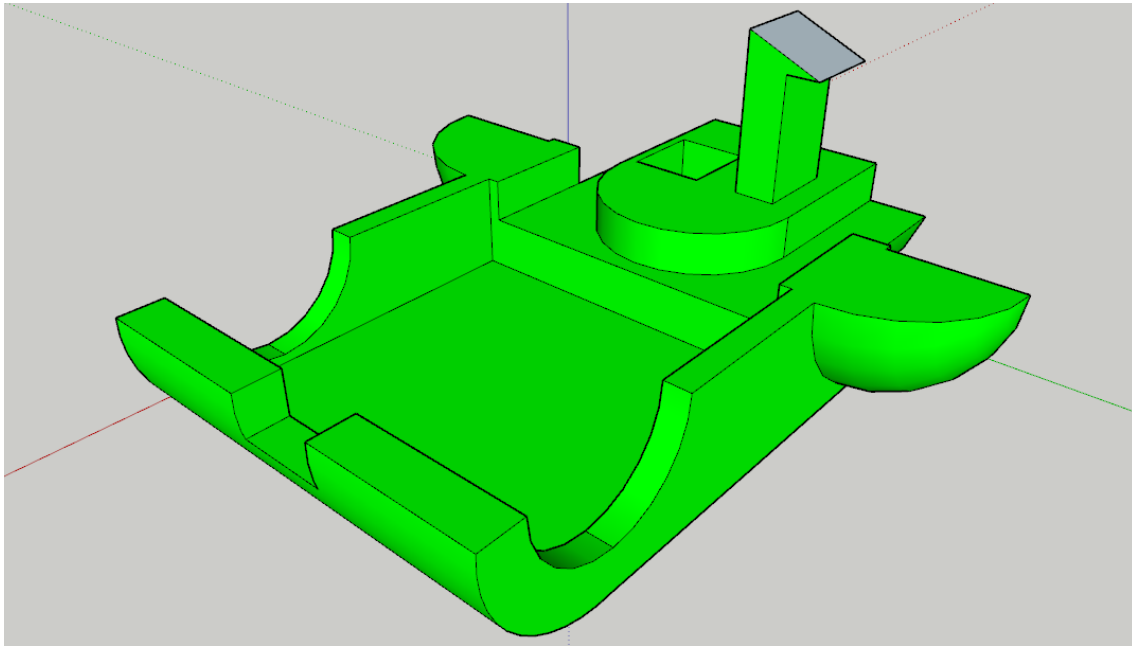


Figura 16. Base.

Diseño inicial de la base, sobre la cual se asentará todo el peso del brazo y conferirá al conjunto un movimiento rotatorio de 180°. Se escogió el sistema de clips también para esta pieza.

Todos los diseños aquí expuestos tienen carácter de prototipos, se ha decidido adjuntarlos al proyecto para mostrar la evolución del mismo.

4– Presentación y Descripción

Antes de desarrollar el funcionamiento proyecto, se ha visto oportuno explicar el funcionamiento por el cual el brazo robótico está provisto de movilidad en todas sus articulaciones.

4.1. PWM (*Pulse Width Modulation*)

Una señal de modulación de ancho pulso (PWM) es un método para generar una señal analógica utilizando una fuente digital. Una señal PWM consta de dos componentes principales que definen su comportamiento: un ciclo de trabajo y una frecuencia. El ciclo de trabajo describe la cantidad de tiempo que la señal está en un estado alto (encendido) como un porcentaje del tiempo total que se tarda en completar un ciclo. La frecuencia determina que tan rápido el PWM completa un ciclo (es decir, 1000 Hz serían 1000 ciclos por segundo) y, por lo tanto, qué tan rápido cambia entre los estados alto y bajo. Al apagar y encender una señal digital a una velocidad suficientemente rápida, y con un cierto ciclo de trabajo, la salida parecerá comportarse como una señal analógica de voltaje constante cuando se suministra energía a los dispositivos.

Gracias al PWM podemos posicionar el servomotor en la posición angular que nosotros deseemos, controlando esta en nuestro caso mediante potenciómetros

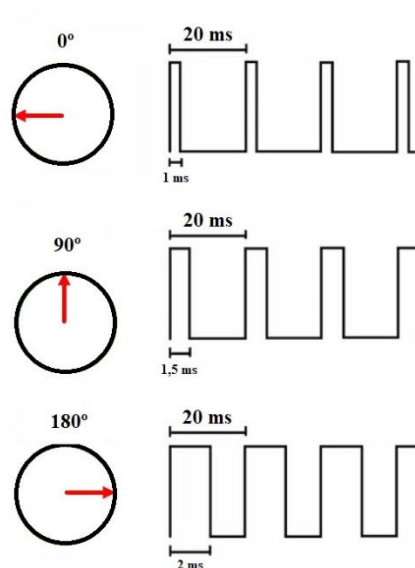


Figura 17: Pulse Width Modulation (Modulación de Ancho de Pulso)

El sistema se basa en una serie de articulaciones que le confieren al conjunto la capacidad de pivotar sobre sí mismo en un ángulo de 180° desde su base. En conjunto, el proyecto consta de 2 estructuras a modo de brazos articulados. El primer brazo, formado por la inclusión de potenciómetros en las articulaciones, recogerá las posiciones en las que debe desplazarse el segundo brazo, de mayor tamaño y formado por servomotores que conforman un sistema de articulaciones.

El modelo de potenciómetros se resolvió diseñándolo con una medida menor, ya que los potenciómetros no pueden aguantar el mismo peso que los servomotores, así ganaríamos versatilidad de cara a su manipulado. Se decidió hacer dos diseños diferenciados, por trabajar un poco más el área de diseño y poder ganar soltura con el software de edición 3d.

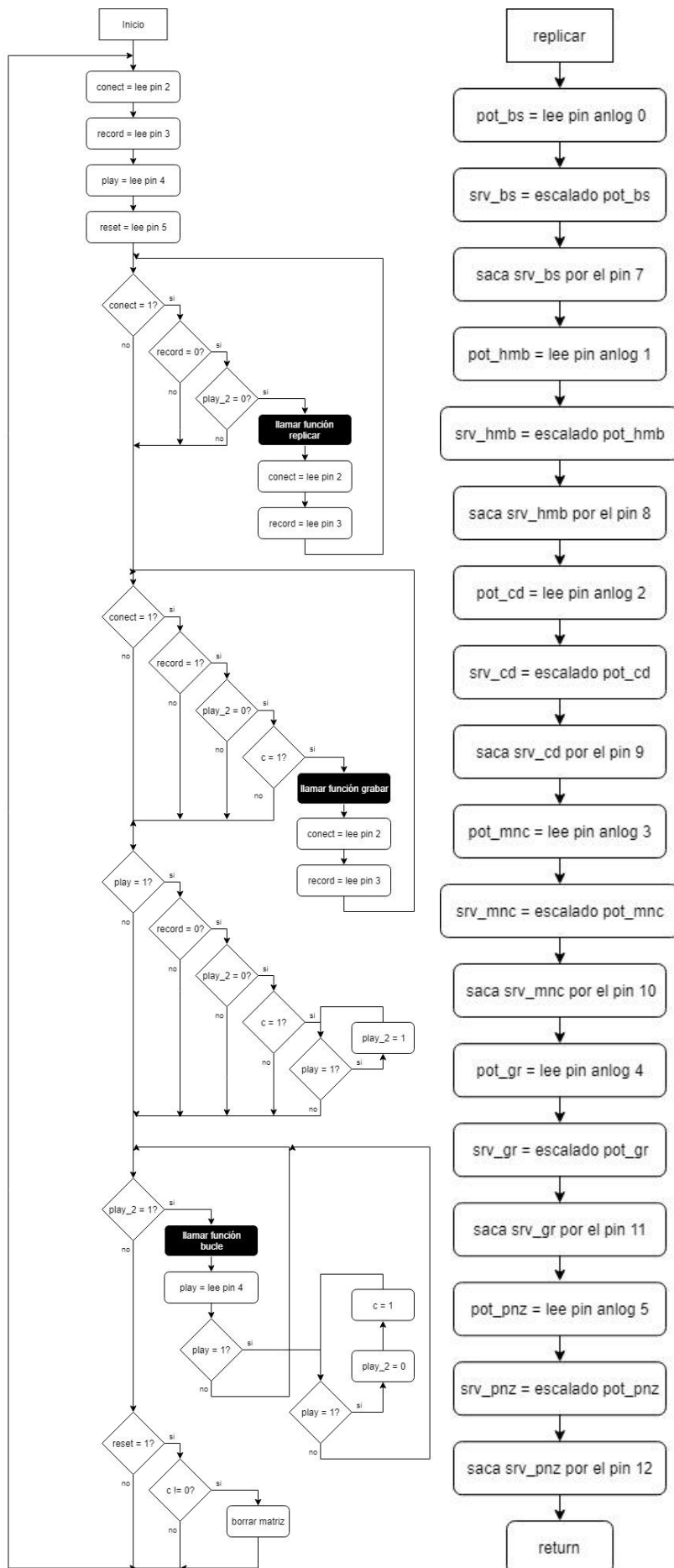
Mediante la placa de desarrollo Arduino y gracias al código, conseguimos que la resistencia que ejercen los potenciómetros en valor analógico, sea traducida gracias al Arduino en “ángulos” o posiciones que los servomotores deben de obedecer, aquí es donde entra el juego el PWM.

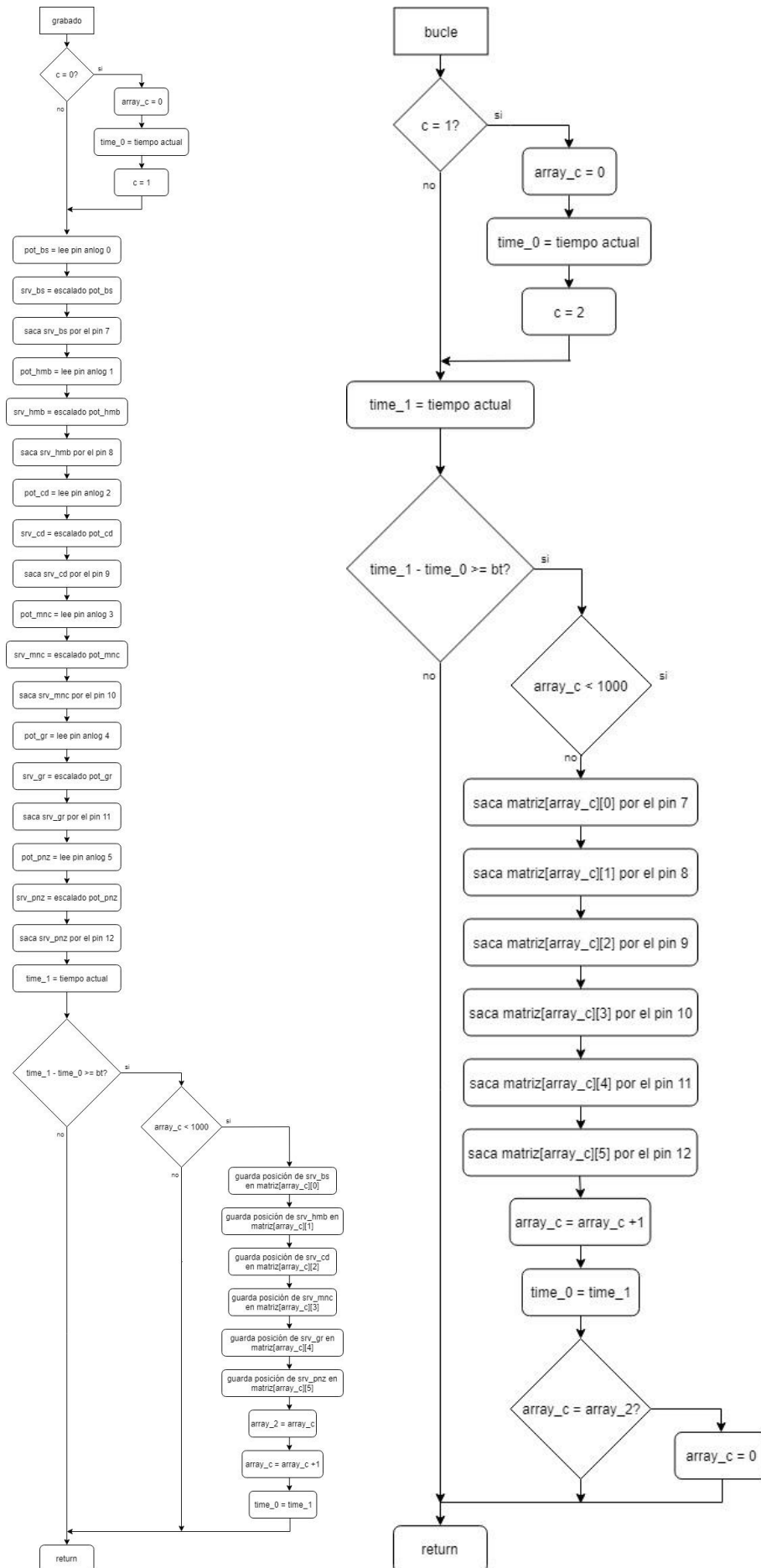
Mediante botonera accionaremos diferentes partes del código que contienen los modos de trabajo, que son: grabación de puntos, replicación de secuencia y reset. Para dar paso al primer modo (grabación) se deberá de accionar el 1º botón de forma continuada mientras se va ejerciendo el movimiento a realizar. Una vez clara la secuencia se soltará dicho botón, y simplemente se accionará el segundo, que da paso al movimiento automático del brazo formado por los servos, replicando así toda la secuencia grabada.

En resumen, al ejercer un movimiento en el primer brazo formado por potenciómetros en sus articulaciones, deberá ser replicado en relación 1:1 por el segundo brazo, constituido por servomotores. Las articulaciones oscilaran en ángulos comprendidos en 180°, siendo el movimiento de los servomotores solidario al de los potenciómetros.

4.2 Diagrama de Flujo

A continuación, se adjunta el diagrama de flujo del funcionamiento del sistema, lo que facilitará al lector la comprensión del conjunto y la posible replicación del proyecto si procede:





5 – Proceso de impresión 3D

El software utilizado para diseño 3D ha sido “SketchUP Pro 2019”

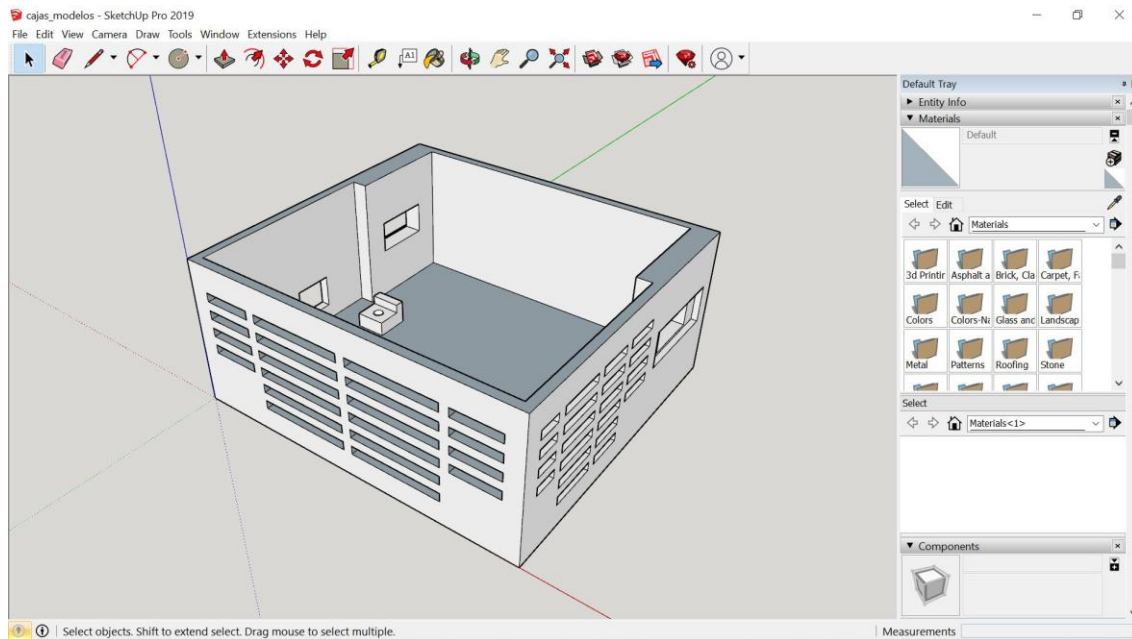


Figura 18: Software de diseño 3D “SketchUP Pro 2019”

Una vez generado el diseño deseado, debemos de exportar el diseño al tipo de formato “.stl”. Este formato es el que acepta la impresora 3D para comenzar a imprimir el modelo.

Durante todo el proyecto tuvimos numerosos problemas de impresión por algún tipo de incompatibilidad entre el software de diseño y el programa de impresión. Después de realizar una búsqueda exhaustiva para tratar de frenar esta falla, se procedió a utilizar un software de apoyo llamado “Slic3r”, con el cual minimizábamos los problemas que surgían en la impresión.

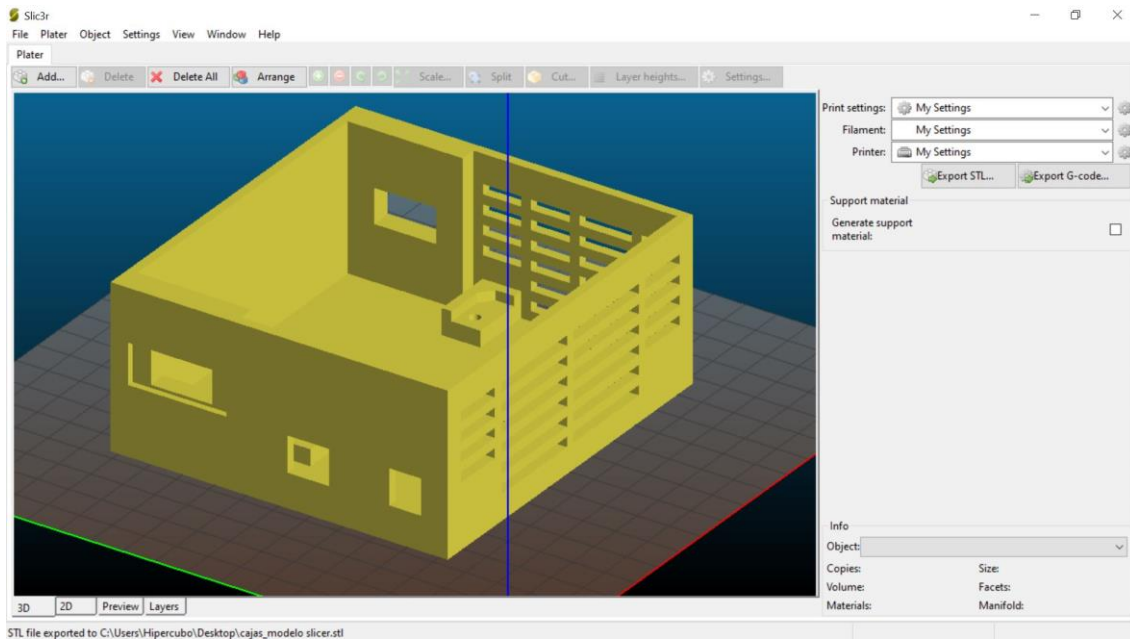


Figura 19: Software de depuración diseño 3D “Slic3r”

Simplemente añadir el archivo con la extensión “.stl” y volver a exportarlo. Con ello conseguimos un diseño más depurado y sin anomalías en la estructura.

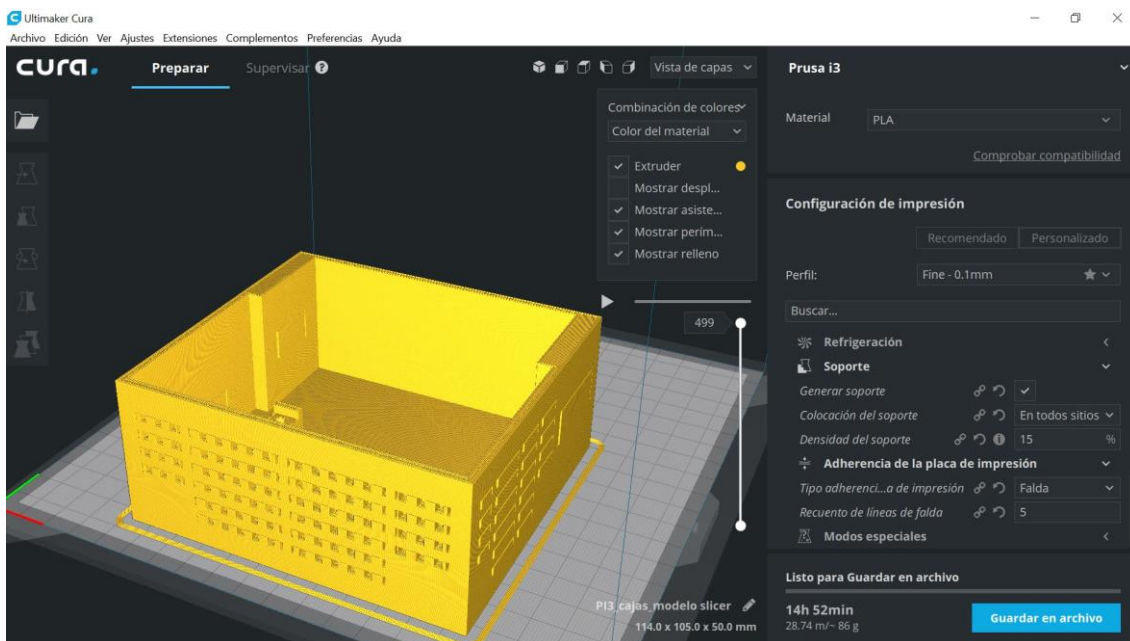


Figura 20: Software de impresión 3D “Ultimaker Cura 3.2.1”

Finalmente, mediante el software “Ultimaker Cura” se configura las características de la impresión (velocidad de barrido, generación de soporte, temperatura del material...).

Una vez más encontramos errores de impresión con la última versión del software, teniendo que recurrir a otro periodo de búsqueda y proceso de ensayo y error para encontrar la solución. Esta no era otra que utilizar la versión del software “Ultimaker Cura 3.2.1”

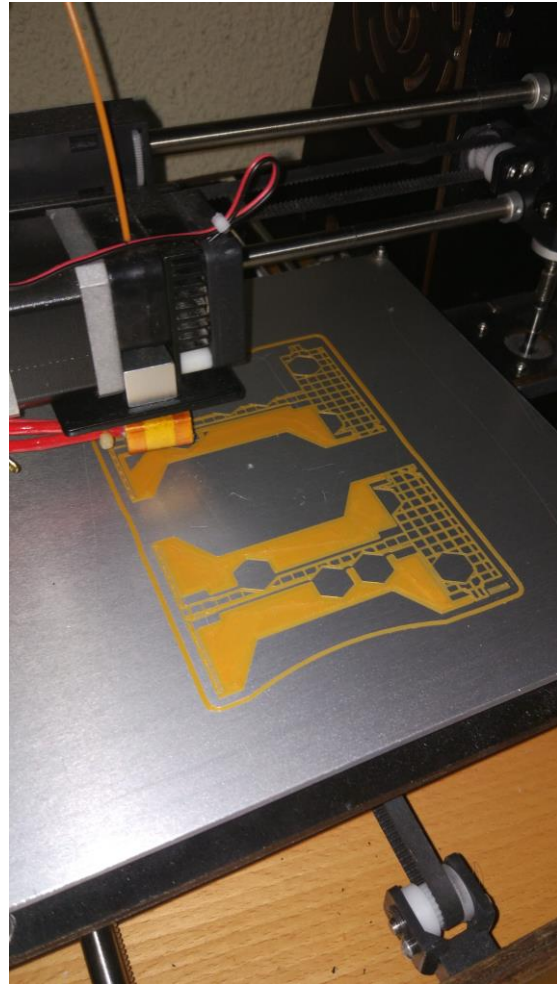
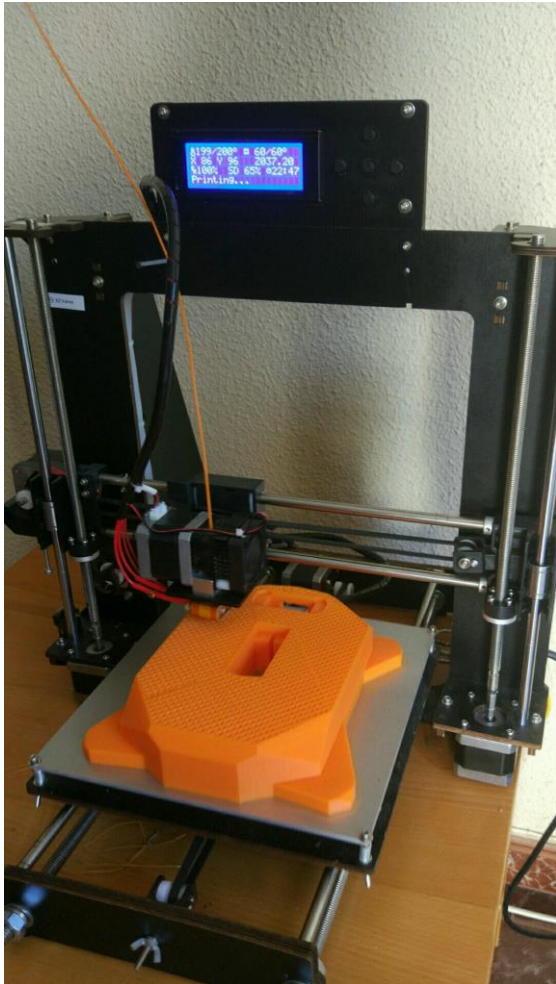


Figura 21: Proceso de impresión 3D

6 – Presupuesto

MATERIAL	PRECIO (€)
Set Conectores Servomotores (x12)	10,99 (x1 ud.)
Cableado	Reciclado
Arduino Mega-2560 R3 (set)	49,99 (x1 ud.)
Fuente de Alimentación	Reciclado
Fundas Termoretráctiles	Reciclado
Soldador	19,95 (x1 ud.)
Estaño	Reciclado
Varilla + Tornillos	0,88 (x1 ud. + x10 uds)
Tornillos + Tuercas + Llaves	8,40 (x1 ud.)
1 kg filamento 3d	20,98 (x1 ud.)
Conectores VGA	0,73 (x6 uds.)
Cable VGA	1,00 (x2 uds.)
Servomotor Futaba S3003	Reciclado
Servomotor SG90	1,69 (x3 uds.)
Servomotor RDS3115MG	9,88 (x3 uds.)
Potenciómetro WH148 B10K	0,96 (x5 uds.)
Piezas extra servomotores	19,98 (x1 uds)
	Total: 185.01 €

Figura 22: Presupuesto

Los servomotores SG90 fueron escogidos pensando en la movilidad y apertura de la herramienta, debido a su pequeño tamaño y funcionalidad, ya que puede ser acoplado en un reducido espacio y cumplir con su cometido.

Los servomotores RDS3115MG, de mayor tamaño que los SG90, tienen por el contrario ejercer la función de articulación del brazo robótico, por lo que decidimos optar por una opción que ejerciese una mayor fuerza, ya que va a soportar mayor peso.

Con respecto a los potenciómetros no se tuvo ninguna consideración en cuenta, salvo que cumplierse con las vueltas relativas a los ángulos de movimiento de los propios servomotores.

7- Planos

Se adjunta plano de cableado donde se muestra la distribución de entradas y salidas del sistema.

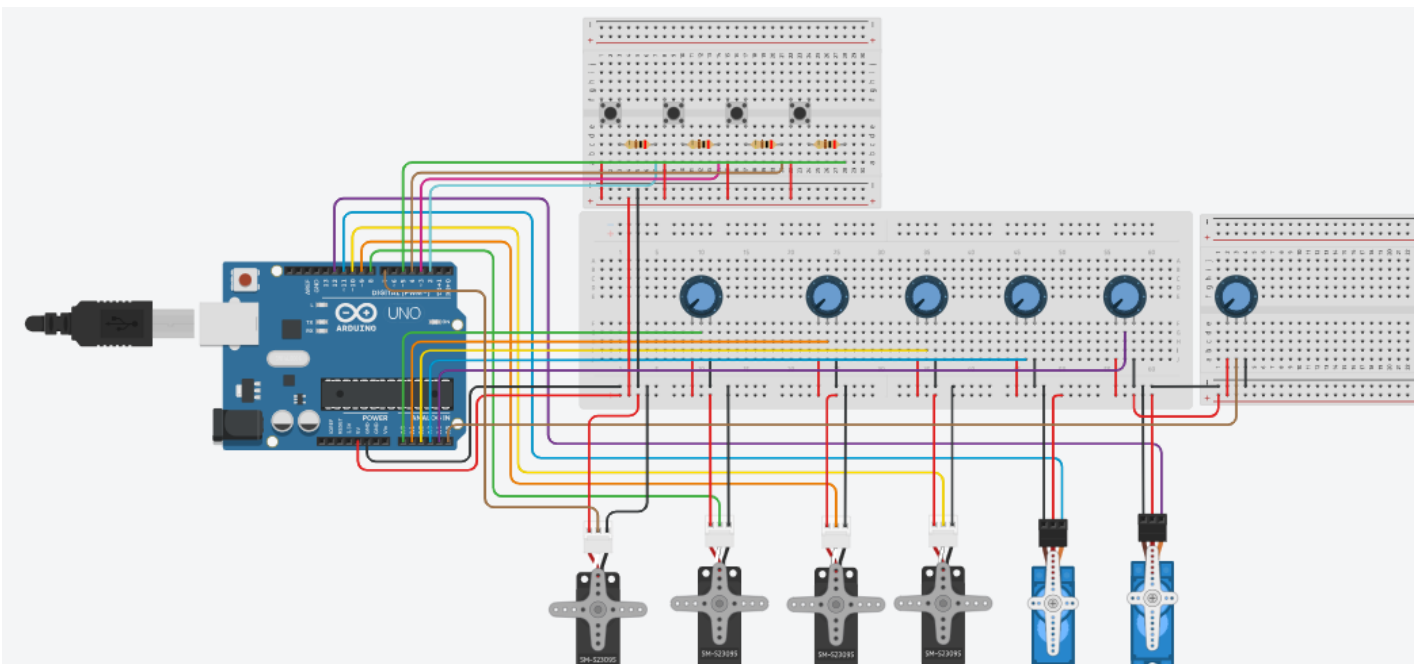


Figura 23: Cableado I/O

Se adjunta diseño final del proyecto previo a la impresión 3D:

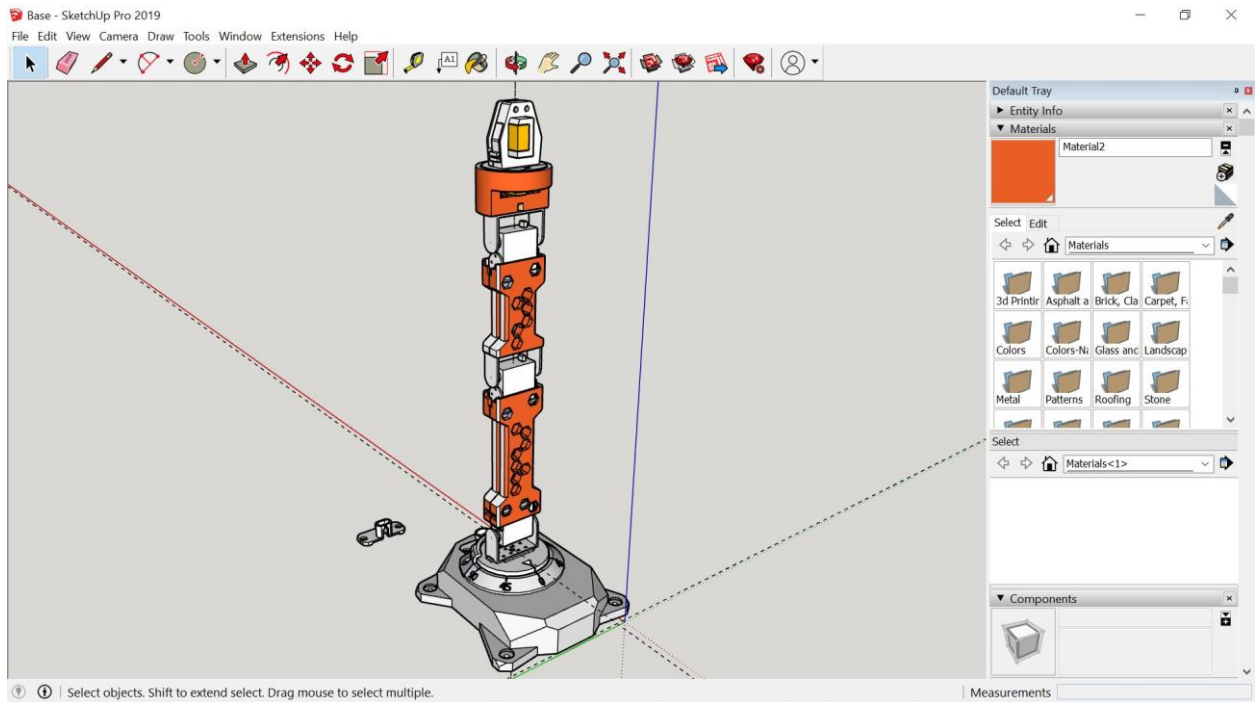


Figura 24: Diseño Final

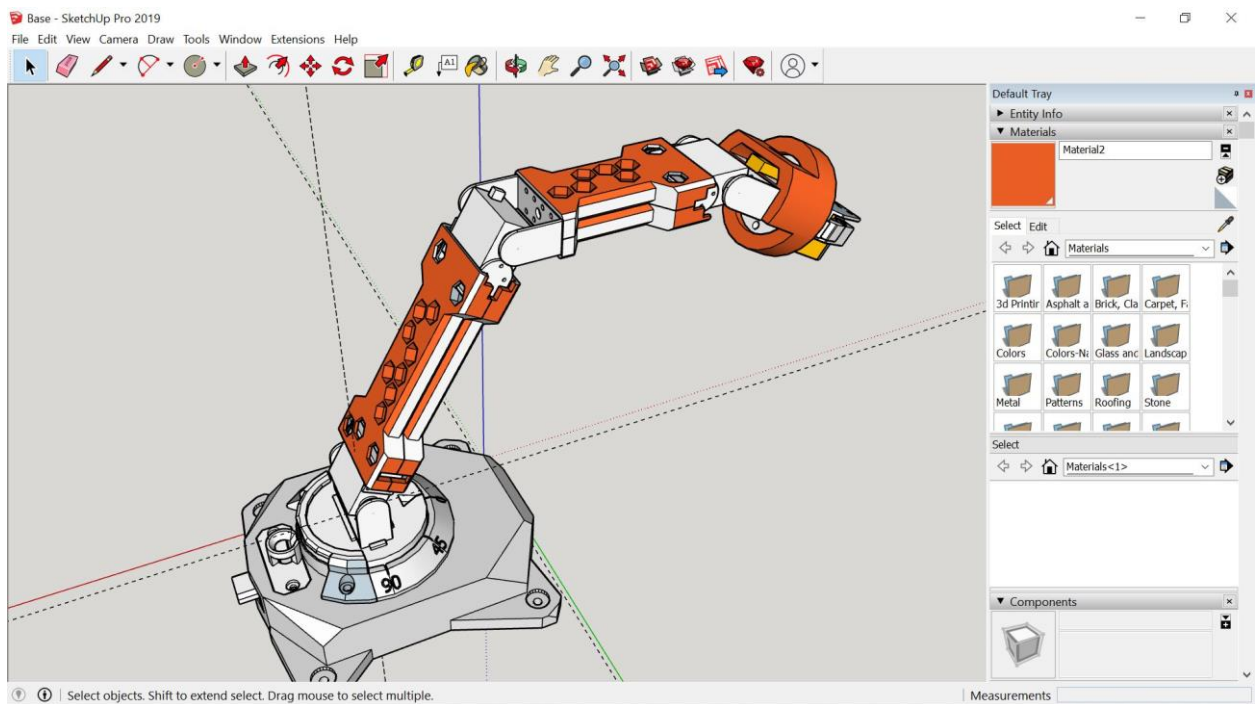


Figura 25: Diseño Final

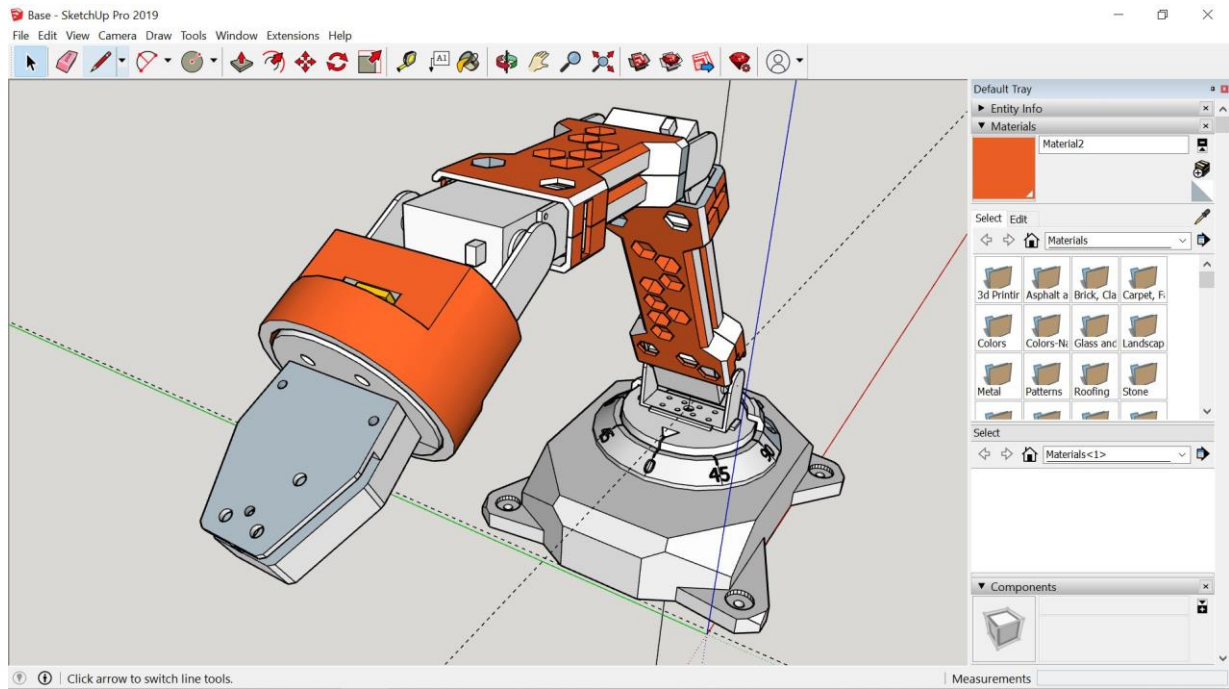


Figura 26: Diseño Final

8- Contratiempos

Los problemas fundamentales a la hora de desarrollar el proyecto han sido afrontar el diseño de las piezas y la impresión de las mismas. El proyecto fue planteado desde un principio como un desafío mutuo, algo que supusiese afrontar el aprendizaje de nuevas habilidades durante el proceso. Es por ello por lo que se decidió añadir el componente de diseño 3D como eje central.

Se tuvieron que afrontar diversos problemas en el apartado de diseño e impresión 3D, ya que ambos éramos profanos en la materia. Se plantearon varios prototipos, se llegaron a imprimir partes de ellos, pudiendo observar problemas no contemplados en el software de diseño que nos limitaban movilidad y funcionalidad de todo el conjunto.

Una vez localizadas las fallas de diseño, se procedió a simplificar el diseño global para que fuese más viable a la hora de la impresión 3D, punto crítico del proyecto, ya que había que tener en cuenta las limitaciones de la propia impresora.

La propia impresión de las piezas nos supuso un hándicap a la hora de avanzar en el proyecto. El depender de una impresora ajena nos retrasó varias semanas,

teniendo que hacer modificaciones en el diseño. A raíz de ello, y una vez el compañero Rafael Antonio Quesada Calahorro dio por finalizado su proyecto, nos cedió su impresora. Ahora bien, a dos semanas de la fecha final de entrega, por lo que se tuvo que trabajar a contrarreloj y aprendiendo sobre la marcha todo lo posible para llegar a tiempo.

A continuación, se adjuntan fotos relacionadas con los fallos de impresión anteriormente mencionados. Un pequeño ejemplo de la problemática:



Figura 26: Fallos de impresión del Modelo 2.2

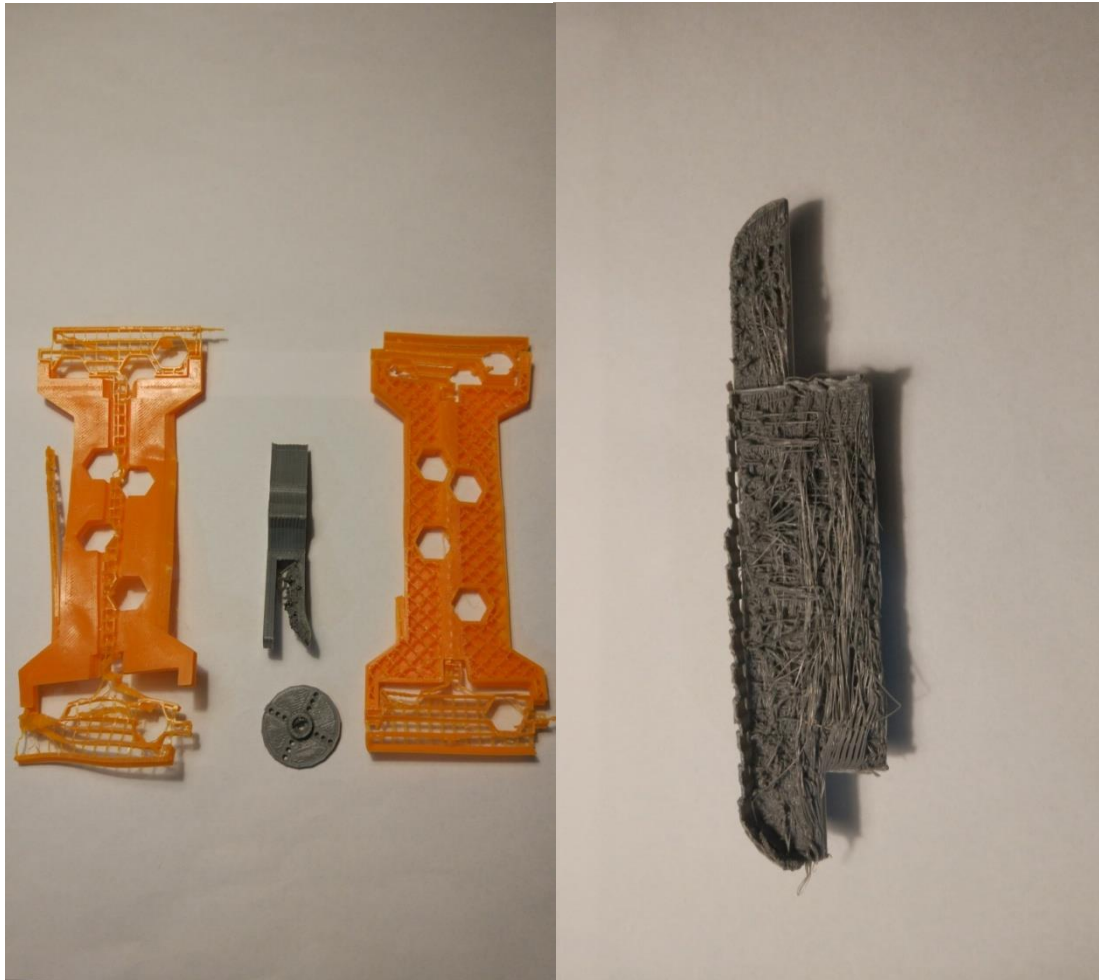


Figura 27: Fallos de impresión brazo/pinza

9- Bibliografía

- Craig, J. (2003). "*Introduction to Robotics. Mechanics & Control*"
- Davies, R. (2015). "*Industria 4.0. Digitalisation for productivity and growth*". European Parliamentary Research Service.
- Del Val Román, J.L. (2016). "*Industria 4.0: la transformación digital de la industria*". Conferencia de Directores y Decanos de Ingeniería Informática (CDDi).
- Comisión Europea (2014). "*Una nueva Revolución Industrial*". Comprender las políticas de la Unión Europea, Oficina de Publicaciones.
- López Ramírez, M (2017). "*50 Ejercicios prácticos de Automatización*" Editorial Marcombo.
- Miranda Peña, F. (2016). "*Características morfológicas. Principales características de los Robots*".
- Romeo, A. (2013). "*Morfología de los robots industriales*"
- Lagunilla. M. (1996). "*Industrial Control and Instrumentation*". Editorial Paraninfo.

ANEXO i

- Tabla de I/O

N.º de pin	Tipo	Función	Conexión	Variable
0	TX0	-	-	-
1	RX0	-	-	-
2	digital_PWM	input	replicador conectado	conect
3	digital_PWM	input	grabar	record
4	digital_PWM	input	reproducir/parar	play
5	digital_PWM	input	reset	reset
6	digital_PWM	-	-	-
7	digital_PWM	output	servo base	srv_bs
8	digital_PWM	output	servo hombro	srv_hmb
9	digital_PWM	output	servo codo	srv_cd
10	digital_PWM	output	servo muñeca	srv_mnc
11	digital_PWM	output	servo giro	srv_gr
12	digital_PWM	output	servo pinza	srv_pnz
13	digital_PWM	-	LED_placa	

N.º de pin	Tipo	Función	Conexión	Variable
A0	analog	input	pot base	pot_bs
A1	analog	input	pot hombro	pot_hmb
A2	analog	input	pot codo	pot_cd
A3	analog	input	pot muñeca	pot_mnc
A4	analog	input	pot giro	pot_gr
A5	analog	input	pot pinza	pot_pnz

- Código

```
#include <Servo.h>
#include <stdio.h>

bool      conect, record, play, reset;

Servo     srv_0,    srv_1,    srv_2,    srv_3,    srv_4,
srv_5;    //añadir servo para su uso
int       pot_bs,   pot_hmb,   pot_cd,   pot_mnc,   pot_gr,
pot_pnz;  //valores de los pote
byte      srv_bs,   srv_hmb,   srv_cd,   srv_mnc,   srv_gr,
srv_pnz;  //valores de los servo

byte      play_2;
byte      array_c, array_2, c;
byte      matriz[1000][5];
unsigned long  time0, time1;

byte      bt = 0;           //en milisegundos
byte      toler = 5;        //tolerancia de guardado de datos +-x
ms

void setup() {

    Serial.begin(9600);

    pinMode(2, INPUT);      //replicaador conectado
    pinMode(3, INPUT);      //grabar
    pinMode(4, INPUT);      //reproducir parar
    pinMode(5, INPUT);      //reset

    srv_0.attach(7);        //servo base  us max y min
    srv_1.attach(8);        //servo hombro
    srv_2.attach(9,70,230);  //servo codo    500-2500
    srv_3.attach(10,70,230); //servo muneca
    srv_4.attach(11,70,230); //servo giro
    srv_5.attach(12);        //servo pinza

}

void loop() {

    conect = digitalRead(2);
    record = digitalRead(3);
    play   = digitalRead(4);
    reset  = digitalRead(5);

    //_____llamar
    replicador_____

    while(conect == HIGH && record == LOW && play_2 == 0){
        //robo_pote conectado

        //puls_grabrar no pulsado
        replicar();                //no esta
        reproduciendo
    }
```

```

    conect  = digitalRead(2);
    record  = digitalRead(3);
}

//_____llamar
grabado_____

    while(conect == HIGH && record == HIGH && play_2 == 0 && c == 0){
//robo_pote conectado

//puls_grabrar pulsado
    grabado();
//no esta reproduciendo

//contador c = 0
    conect  = digitalRead(2);
    record  = digitalRead(3);
}

//_____llamar
bucle_____

    if(play == HIGH && record == LOW && play_2 == 0 && c == 1){

        while(play == HIGH){

            play_2 = 1;
        }
    }

    while(play_2 == 1){

        bucle();

        record  = digitalRead(3);
        play    = digitalRead(4);

        if(play == HIGH && record == LOW){

            while(play == HIGH){

                play_2 = 0;
                c = 1;
            }
        }
    }

//_____reset_____

    if(reset == HIGH && c != 0){

        for(array_c = 1000; array_c >= 0; array_c --){
            matriz[array_c][0] = 0;
            matriz[array_c][1] = 0;
            matriz[array_c][2] = 0;
            matriz[array_c][3] = 0;
            matriz[array_c][4] = 0;
            matriz[array_c][5] = 0;

```

```

        c = 0;
    }
}

}

// _____replicar_____
// _____

void replicar() {

    //base
    pot_bs = analogRead(0);           //lee pote
    srv_bs = map(pot_bs, 0, 1023, 8, 188); //escalado
    srv_0.write(srv_bs);              //mueve valor al
    servo

    //hombro
    pot_hmb = analogRead(1);          //lee pote
    srv_hmb = map(pot_hmb, 0, 1023, 0, 180); //escalado
    srv_1.write(srv_hmb);             //mueve valor al
    servo

    //codo
    pot_cd = analogRead(2);           //lee pote
    srv_cd = map(pot_cd, 0, 1023, 0, 180); //escalado
    srv_2.write(srv_cd);             //mueve valor al
    servo

    //muneca
    pot_mnc = analogRead(3);          //lee pote
    srv_mnc = map(pot_mnc, 0, 1023, 0, 180); //escalado
    srv_3.write(srv_mnc);            //mueve valor al
    servo

    //giro
    pot_gr = analogRead(4);           //lee pote
    srv_gr = map(pot_gr, 0, 1023, 0, 180); //escalado
    srv_4.write(srv_gr);             //mueve valor al
    servo

    //pinza
    pot_pnz = analogRead(5);          //lee pote
    srv_pnz = map(pot_pnz, 0, 1023, 0, 180); //escalado
    srv_5.write(srv_pnz);            //mueve valor al
    servo
}

// _____grabado_____
// _____

void grabado() {

    if(c == 0){

        array_c = 0;
        time0 = millis();
        c = 1;
    }
}

```

```

    }

    //base
    pot_bs = analogRead(0); //lee pote
    srv_bs = map(pot_bs, 0, 1023, 8, 188); //escalado
    srv_0.write(srv_bs); //mueve valor al
servo

    //hombro
    pot_hmb = analogRead(1); //lee pote
    srv_hmb = map(pot_hmb, 0, 1023, 0, 180); //escalado
    srv_1.write(srv_hmb); //mueve valor al
servo

    //codo
    pot_cd = analogRead(2); //lee pote
    srv_cd = map(pot_cd, 0, 1023, 0, 180); //escalado
    srv_2.write(srv_cd); //mueve valor al
servo

    //muneca
    pot_mnc = analogRead(3); //lee pote
    srv_mnc = map(pot_mnc, 0, 1023, 0, 180); //escalado
    srv_3.write(srv_mnc); //mueve valor al
servo

    //giro
    pot_gr = analogRead(4); //lee pote
    srv_gr = map(pot_gr, 0, 1023, 0, 180); //escalado
    srv_4.write(srv_gr); //mueve valor al
servo

    //pinza
    pot_pnz = analogRead(5); //lee pote
    srv_pnz = map(pot_pnz, 0, 1023, 0, 180); //escalado
    srv_5.write(srv_pnz); //mueve valor al
servo

    time1 = millis();

    if((time1 - time0) > (bt - toler) && (bt + toler) < (time1 -
time0) && array_c < 1000){ //guardar datos cada 'bt' (+-5ms) ms

        time0 = time1;

        matriz[array_c][0] = srv_bs;
        matriz[array_c][1] = srv_hmb;
        matriz[array_c][2] = srv_cd;
        matriz[array_c][3] = srv_mnc;
        matriz[array_c][4] = srv_gr;
        matriz[array_c][5] = srv_pnz;

        array_2 = array_c;
        array_c = array_c + 1;
    }
}

```

```

// _____bucle_____
_____

void bucle(){

    if(c == 1){

        time0 = millis();
        c = 2;
        array_c = 0;
    }

    time1 = millis();

    if((time1 - time0) >= bt && array_c <= array_2){

        srv_0.write(matriz[array_c][0]);
        srv_1.write(matriz[array_c][1]);
        srv_2.write(matriz[array_c][2]);
        srv_3.write(matriz[array_c][3]);
        srv_4.write(matriz[array_c][4]);
        srv_5.write(matriz[array_c][5]);

        array_c = array_c + 1;
        time0 = time1;

        if(array_c == array_2){

            array_c = 0;
        }
    }
}

```

ANEXO ii

- Planos Acotados

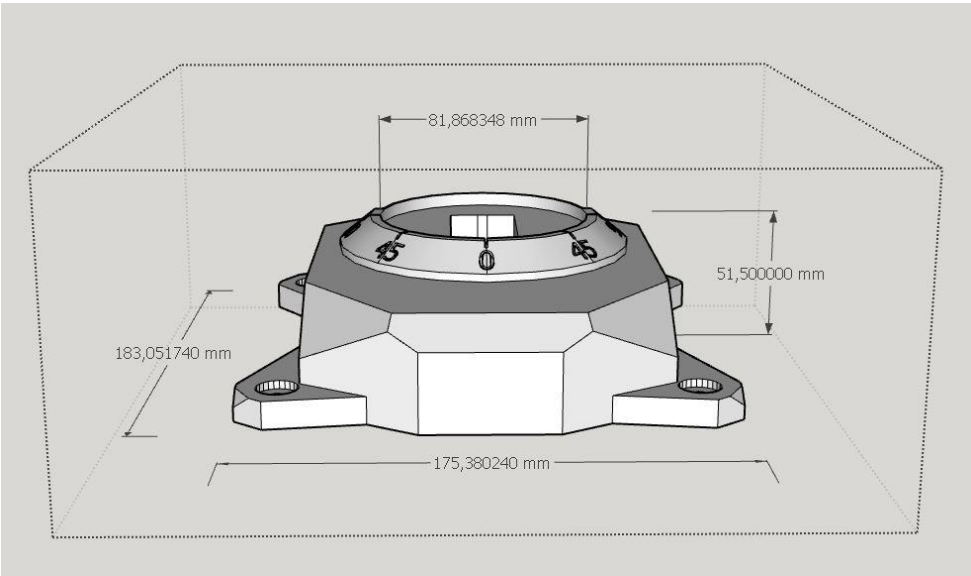


Figura 28: Cotas Base

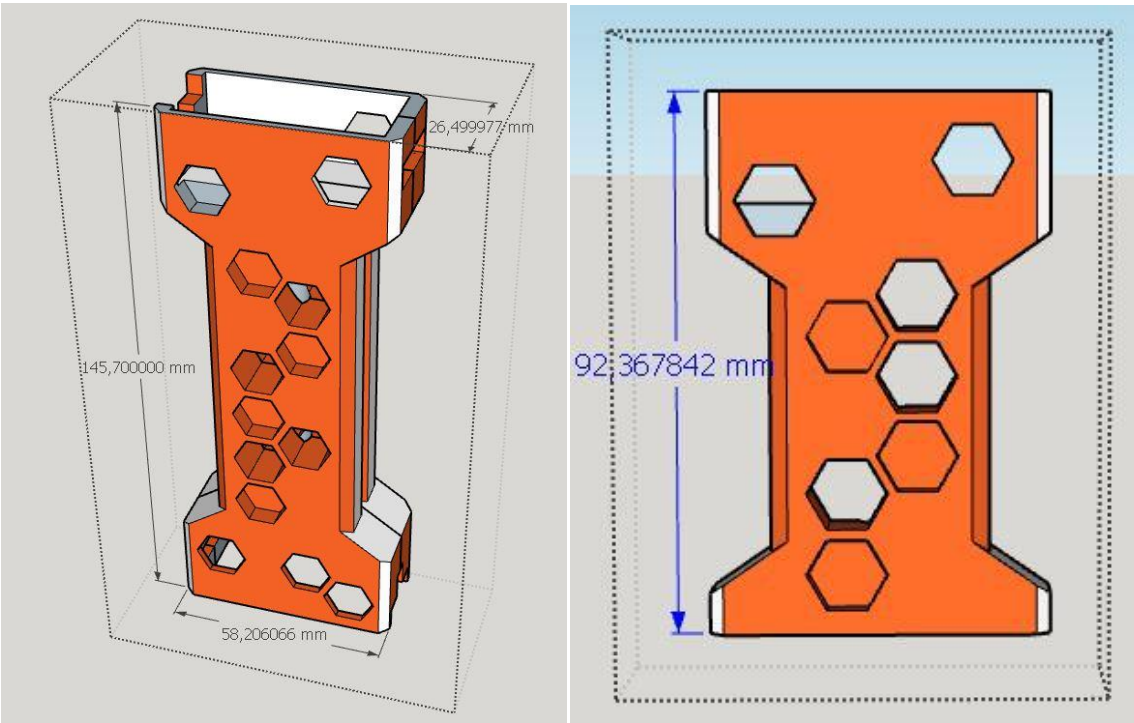


Figura 29: Cotas brazo

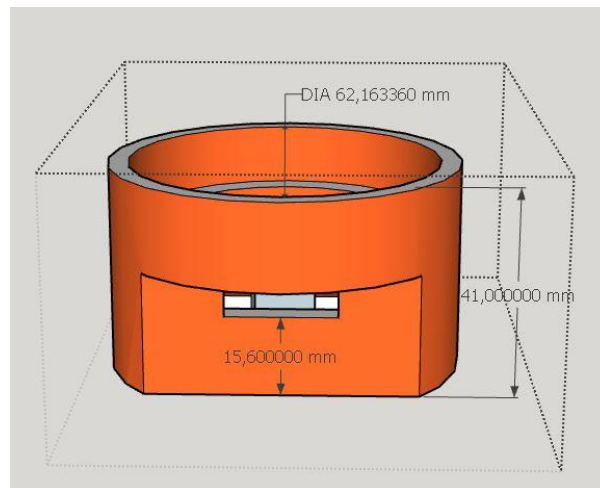


Figura 29: Cotas Brida

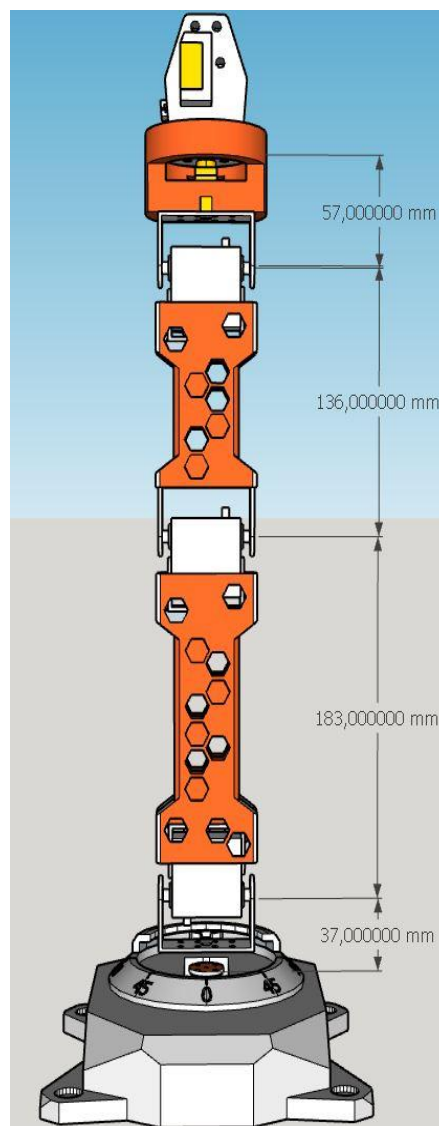


Figura 30: Cotas entre ejes